# CS 255 Business Requirements Document Template

Complete this template by replacing the bracketed text with the relevant information.

This template lays out all the different sections that you need to complete for Project One. Each section has guiding questions to prompt your thinking. These questions are meant to guide your initial responses to each area. You are encouraged to go beyond these questions using what you have learned in your readings. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length for the final document. Instead, the goal is to complete each section based on your client’s needs.

**Tip:** You should respond in a bulleted list for each section. This will make your thoughts easier to reference when you move into the design phase for Project Two. One starter bullet has been provided for you in each section, but you will need to add more.

## System Components and Design

### Purpose

*What is the purpose of this project? Who is the client and what do they want their system to be able to do?*

* Liam wants to build a system that helps people with passing their driver test.
* Liam has been noticing that more people are failing their driving test.
* He wants to create a system that people can get access to and help with passing their driving test.
* The client is DriverPass, a new business aiming to offer both online and on-the-road training.
* The goal is to create a cloud-based system that supports student registration, appointment scheduling, progress tracking, and administrative functions for internal staff.

### System Background

*What does DriverPass want the system to do? What is the problem they want to fix? What are the different components needed for this system?*

* DriverPass wants the system to be able to allow people access to a database that can help with their driving test.
* The problem that DriverPass wants to fix is the problem of too many people failing their driving test.
* The system needs to offer online driving lessons and DMV rule content.
* Practice exams with tracked results.
* On-the-road training appointments with matched cars and instructors.
* A scheduling system for booking and modifying lessons.
* Role-based access for admin, IT, secretary, and students.
* Reports and activity logs.
* Integration with DMV systems for updates.
* Mobile and desktop compatibility.

### Objectives and Goals

*What should this system be able to do when it is completed? What measurable tasks need to be included in the system design to achieve this?*

* This system, when it is complete, should allow users to access a database that helps them pass their driving test.
* Users should have the ability to choose from three specific packages and create and modify their appointments as needed.
* An admin should have full control over the system. Has the ability to access it online and offline, with the ability to edit system data. Download reports and information to work at home.
* Systems should have a measure for secure password and login functionality for users.
* A functioning table where you can see driver notes and an input function where the student (or secretary) enters the students’ info.
* It should be cloud based so the system can be accessed from all internet devices.
* The system should allow IT admin to access accounts for login and password recovery.
* Have access and be notified from the DMV of new updates to users and driving instructors.
* The system should have a table or list that shows the progress of tests taken or completed by the customer.
* The system should include a contact page for student inquiries.

## Requirements

### Nonfunctional Requirements

*In this section, you will detail the different nonfunctional requirements for the DriverPass system. You will need to think about the different things that the system needs to function properly.*

#### Performance Requirements

*What environments (web-based, application, etc.) does this system need to run in? How fast should the system run? How often should the system be updated?*

* This system should be accessible on both desktop and mobile devices, supporting common browsers (e.g. Chrome, Firefox, Edge, Safari) and mobile platforms (e.g. iOS and Android).
* Pages should load in under 3 seconds to keep user engagement and reduce frustration user frustration.
* The application should be able to run under a load of 1,000 concurrent users without significant degradation in performance.
* Regular system updates should be scheduled biweekly to patch bugs and monthly to add features, with zero downtime during peak usage hours.

#### Platform Constraints

*What platforms (Windows, Unix, etc.) should the system run on? Does the back end require any tools, such as a database, to support this application?*

* The system must be compatible with major operating systems, including Windows and Linux and must function on mobile operating systems such as Android and iOS.
* The back end needs to utilize a cloud-based relational database to support scalable storage of user information, progress tracking, and appointment data.
* The system must be browser-compatible, functioning across all major modern browsers.
* The system must be able to integrate with DMV systems in order to stay up to date to enable real-time updates whenever policy changes have been made.

#### Accuracy and Precision

*How will you distinguish between different users?* *Is the input case-sensitive? When should the system inform the admin of a problem?*

* The system shall distinguish between users based on their role type (admin, IT, secretary, student) using a unique email and secure password combination for authentication.
* All input fields shall enforce case sensitivity, proper format validation (e.g., for email, phone numbers), and appropriate character limits to ensure clean and predictable data.
* The system shall automatically alert administrators if invalid input data is repeatedly entered (e.g., multiple failed form submissions) or if anomalous behavior is detected (e.g., repeated failed login attempts suggesting a brute-force attack).

#### Adaptability

*Can you make changes to the user (add/remove/modify) without changing code? How will the system adapt to platform updates? What type of access does the IT admin need?*

* The system shall allow administrators to access and manage all user data, including the ability to reset passwords, deactivate accounts, or block access to ensure secure and flexible account control.
* The system shall support dynamic updates to content and policies, allowing for modification of DMV regulations or training material without requiring a codebase change.
* IT should have full administrative access to perform system maintenance, apply security patches, resolve bugs, and update system components as needed with minimal downtime.

#### Security

*What is required for the user to log in? How can you secure the connection or the data exchange between the client and the server? What should happen to the account if there is a “brute force” hacking attempt? What happens if the user forgets their password?*

* The system will require users to use a valid email and secure password that meets minimum complexity requirements to login.
* The system should have the ability to monitor any brute force attempts to login and automatically notify admins after a threshold of failed attempts, while also temporarily locking the affected account.
* All data exchange from client to server should be using industry-standard encryption protocols, as in HTTPS and TLS, to protect sensitive data.
* In the event an user forgets their password, the system should have an option to where they can choose to reset their password via sending them an email to reset it.

### Functional Requirements

*Using the information from the scenario, think about the different functions the system needs to provide. Each of your bullets should start with “The system shall . . .” For example, one functional requirement might be, “The system shall validate user credentials when logging in.”*

* The system shall allow users to register for an account using a valid email and password.
* The system shall allow users students to schedule and modify on-the-road training sessions.
* The system shall allow admins to generate reports on user performance.
* The system shall validate user credentials when logging in.
* The system shall allow admins to modify, update, or delete all system data, including user records, schedules, and progress logs.
* The system shall allow customers to choose a package out of three choices.
* The system shall generate DMV policy update notifications.

### User Interface

*What are the needs of the interface? Who are the different users for this interface? What will each user need to be able to do through the interface? How will the user interact with the interface (mobile, browser, etc.)?*

* All users shall be able to access the system via a web-based interface, accessible from both desktop and mobile browsers for maximum flexibility.
* Students shall have access to an interface that displays their profile information, their schedule, training package details, and test progress with options to book or cancel lessons and securely log out.
* Administrators should have a user interface that enables them to view, add, edit, and delete all system data, including user accounts, appointments, progress reports, and training materials.
* IT personnel have access to a system dashboard to monitor system performance, review logs, reset user credentials, and apply system updates or patches.
* Secretaries or office staff shall have an interface that allows them to manage student records, assist with scheduling, and enter instructor notes following on-the-road sessions.
* The user interface shall follow intuitive navigation standards, including clearly labeled menus, form validation, and responsive design to ensure accessibility across devices.

### Assumptions

*What things were not specifically addressed in your design above? What assumptions are you making in your design about the users or the technology they have?*

* It is assumed that all users have access to a reliable internet connection and a device capable of accessing a web browser.
* It is assumed that the DMV will send timely and accurate updates to policies and procedures through available API integrations.
* It is assumed that a user will enter accurate information when registering for an account and services.
* It is assumed that instructors will log notes and provide updated progress with their students in a timely manner.
* It is assumed that all third-party services will work as expected to protect user’s sensitive information.
* It is assumed that IT and administration will have the necessary training and technical expertise to manage the system’s functionality and respond to user issues effectively.

### Limitations

*Any system you build will naturally have limitations. What limitations do you see in your system design? What limitations do you have as far as resources, time, budget, or technology?*

* The system will likely during the first phase will not have a mobile app, users must access via browser.
* Budget constraints can hinder any other added features to the system other than the core features to run the system.
* Full time DMV policies and procedures updates will likely not be integrated at first launch and may rely on manual updates until full API integration is implemented.
* The system upon release will only have English-only language support and this will limit access to non-English-speaking users.
* Scheduling will likely be reduced to office hours only and may not support same day booking.
* There will be a limited number of staff which will affect 24/7 support upon release.
* In the first initial phase there will be a limited number of instructors, which will limit the flexibility of student scheduling.

### Gantt Chart

*Please include a screenshot of the GANTT chart that you created with Lucidchart. Be sure to check that it meets the plan described by the characters in the interview.*

**